

Universidad Nacional de Mar del Plata

Facultad de Ingeniería

Proyecto: SELFIE

Carrera: Ingeniería Electrónica

PLAN DE PRUEBAS

Alcance

El siguiente plan de pruebas abarca desde las pruebas individuales de los módulos, hasta las pruebas finales de aceptación del instrumento como un conjunto.

Ambientes de prueba

Los ambientes de prueba fueron tres:

- PC.
- PC + Sensor Aptina + FPGA.
- PC + Sensor Aptina + FPGA + Piedra de Americio 241.

Recursos, herramientas e instrumentos

- Placa de desarrollo Zedboard.
- PC.
- Sensor Aptina MTM9001.
- Piedra de Americio 241.
- Contenedor oscuro.
- Lente.

Políticas de trabajo

Las pruebas que se puedan realizar solo con el uso de la PC, se podrán realizar tanto en el Laboratorio de Sistemas Caóticos como en el hogar de los alumnos, comunicándose de por medio.

El resto de ellas serán desarrolladas en las instalaciones del Laboratorio, ya que requieren el uso de la FPGA y el sensor Aptina.

Por otro lado, para aquellas que requieran la manipulación de la piedra de Americio, solo uno de los alumnos manipulará la fuente de radiación y no tocará nada más, mientras que el otro se encargará del manejo de la PC. Cada vez que sea manipulada, el estudiante deberá lavarse las manos de manera exhaustiva sin tocar nada más, para ello el otro alumno abrirá las puertas y la canilla para el lavado. Una vez finalizado el uso de la piedra, deberá guardarse en el recipiente y caja correspondiente.

Estrategia de Comunicación

Los resultados de las pruebas se comunicarán en reuniones con los directores de tesis Maximiliano Antonelli y Claudio Marcelo González, y con los investigadores del Laboratorio de Bajas Temperaturas del Instituto Balseiro, Martín Pérez y José Lipovetzky.

Módulo	ID	Prueba	Tipo de Prueba	Procedimiento	Resultados esperados	Instrumental	Fecha de Prueba	Observaciones	Resultados obtenidos
I2cmaster	1	Escritura y lectura de registros	Unitaria	Leer registro 0x00. Escribir los valores de ganancia 0x01, 0x08 y 0x15 en el registro 0x35. Leer el registro de ganancia una vez que se escribió un valor. Poner en alto el bit 6 del registro 0x07 para que el sensor entregue la imagen de pruebas.	La lectura del registro 0x00 debe devolver el valor 0x8431. El valor escrito en el registro de ganancia debe coincidir con el valor leído. La imagen obtenida con el sensor en modo de pruebas debe estar formada por franjas negras y blancas verticales.	FFGA Vivado	15/02/20	-	Acordé a lo esperado
	2	Comunicación con la FPGa y modos de funcionamiento	Unitaria	Conectarse por puerto UART. Escribir y leer registros. Capturar imágenes con un lente que serán transferidas por DMA a la computadora. Enviar imágenes por GPIO. Crear imágenes e histograma.	Conectarse y desconectarse correctamente de la UART. Realizar nuevamente los procedimientos de la prueba 1 y obtener las salidas indicadas. La imagen capturada coincide con el objeto colocado encima del sensor. La imagen enviada por GPIO y recibida es la misma. En la reconstrucción de imágenes no hay píxeles descorrelacionados, el histograma con bins unitarios coincide con el obtenido en Python y los datos son correctamente agrupados cuando se modifica el binreado.	FFGA Lente Vivado Vivado SDK	15/10/21	-	Acordé a lo esperado
Algoritmo de detección ideal	3	Verificación de detección de eventos	Unitaria	Generar imágenes arbitrarias de 5x5, 10x10 y 25x25. Luego, enviar imágenes de irradiaciones.	La cantidad de eventos y la intensidad de la imagen generada y lo reportado por el algoritmo coinciden.	IDE/Console	10/07/20	-	Acordé a lo esperado
	4	Verificación de detección de eventos	Unitaria	Generar imágenes arbitrarias de 5x5, 10x10 y 25x25. Verificar manualmente si el resultado es correcto. Luego, enviar imágenes de irradiaciones.	La cantidad de eventos y la intensidad de la imagen generada y lo reportado por el algoritmo coinciden.	IDE/Console	20/02/21	-	Acordé a lo esperado
Algoritmo de detección real	5	Verificación de píxeles muertos	Unitaria	Procesar las imágenes de irradiaciones y obtener los píxeles muertos tomando como píxeles muertos a aquellos píxeles que quedan encendidos comparando a la imagen anterior. Luego, procesar las imágenes de irradiaciones tomando como píxeles muertos a aquellos píxeles que quedan encendidos y no tienen otros píxeles encendidos a su alrededor.	La cantidad de píxeles muertos reportados por ambos métodos debe coincidir.	IDE/Console	18/08/21	-	Acordé a lo esperado
	6	Verificación de detección de eventos	Unitaria	Generar imágenes arbitrarias de 5x5, 10x10 y 25x25. Luego, enviar imágenes de irradiaciones, enviándolas en formato serie de 8 bits.	La cantidad de eventos y la intensidad de las imágenes reportadas por el bloque detector deben coincidir con lo reportado por el algoritmo de detección real.	FFGA Vivado	30/08/21	-	Problemas con el reset, las variables static no se reinician con el reset del bloque. Se solucionó con una instrucción pragma.
Detector	7	Tiempo de procesamiento	Unitaria	Enviar imágenes arbitrarias de 5x5, 10x10, 25x25 e imágenes de irradiaciones de 480x720 y 1280x1024. Anotar tiempos de descarte de píxeles por debajo del umbral, tiempos de procesamiento de eventos con varios píxeles y tiempo total. Probar leves modificaciones al HLS y volver a probar iterando.	Los tiempos de procesamiento deben bajar con cada iteración.	FFGA Vivado	24/5/2021 al 16/6/2021	Proceso iterativo.	La herramienta reportaba 20.44 W de consumo de potencia, resultado de un mal ruteo del clock. Una vez arreglado, los resultados fueron acordes a lo esperado.
	8	Generación de histogramas	Unitaria	Generar arrays arbitrarios de tamaño e intensidad y enviarlos en formato serie de 8 bits.	El histograma enviado y el reportado por el bloque deben coincidir.	FFGA Vivado	31/08/21	-	Problemas con el reset, las variables static no se reinician con el reset del bloque. La solución hecha en el detector no es implementable porque ocupa 99% de LUTs, se utilizó un for loop para limpiar las posiciones.
Generador de histograma	9	Verificación de píxeles muertos	Unitaria	Enviar imágenes de irradiaciones.	En las primeras 2*umbral posiciones del histograma debe haber datos de los píxeles muertos.	FFGA Vivado	29/06/21	-	Acordé a lo esperado
	10	Detección de eventos y generación de histograma	Integral	Utilizar las mismas imágenes de las pruebas 4 y 5, enviar los datos de los píxeles en formato serie de 8 bits. Primero enviar de a una imagen, luego enviar de a dos. Enviar imágenes de: 5x5, 10x10, 25x25, 480x720 y 1280x1024	El histograma generado debe coincidir con lo calculado, para las pruebas de imágenes pequeñas, y con lo obtenido mediante el procesamiento en Python, para las imágenes más grandes.	FFGA Vivado	30/06/21	-	Acordé a lo esperado
Self Start (SS)	11	Transición de estados y generación de señales	Unitaria	Enviar las señales y pulsos correspondientes para los cambios de estado.	Los cambios de estado se producen cuando corresponde y las señales de salida tienen los valores correctos.	FFGA Vivado	11/08/21	-	Acordé a lo esperado
	12	Sincronización de datos de entrada	Unitaria	Enviar señales de entrada.	2 ciclos de clocks después debe aparecer el cambio en la salida	FFGA Vivado	09/09/21	-	Acordé a lo esperado
Capture Input Controller (IC)	13	Transición de estados y generación de señales	Unitaria	Enviar las señales y pulsos correspondientes para los cambios de estado. Verificar que estos cambios se producen y que las señales de salida tienen los valores correctos	Los cambios de estado del bloque se producen cuando corresponde y las señales de salida tienen los valores correctos.	FFGA Vivado	26/08/21	-	Acordé a lo esperado
	14	Generación del pulso para escribir la FIFO	Unitaria	Ingresar con un clock a 100 MHz y una señal a 10 MHz.	Se genera un pulso de 10ns (1 clock) cada vez que llega un pulso de frecuencia 10 MHz	FFGA Vivado	25/08/21	-	Acordé a lo esperado
Auto Mode Logic Controller (AMLC)	15	Transición de estados y generación de señales	Unitaria	Enviar las señales y pulsos correspondientes para los cambios de estado.	Los cambios de estado del bloque se producen cuando corresponde y las señales de salida tienen los valores correctos.	FFGA Vivado	11/08/21	-	Acordé a lo esperado
	16	Transición de estados y generación de señales	Integral	Enviar las señales y pulsos correspondientes para los cambios de estado.	Los cambios de estado del bloque se producen cuando corresponde y las señales de salida tienen los valores correctos.	FFGA Vivado	15/10/21	-	Acordé a lo esperado
tvalid_bin	17	Generación de las señales valid_bin, tlast y tdata para el modulo data_io_axi	Unitaria	Enviar 1310720 pulsos a 10 MHz con una frecuencia de clock de 100 MHz.	Verificar que tlast se genere en el último pulso.	FFGA Vivado	02/09/21	-	Acordé a lo esperado
	18	Lectura de datos y generación de señales	Unitaria	Enviar las señales y pulsos correspondientes para los cambios de estado.	Verificar que los cambios de estado se producen, que las señales de salida tienen los valores correctos y que se leen los datos enviados desde el Vivado SDK.	FFGA Vivado Vivado SDK	23/09/21	-	Acordé a lo esperado
SELFIE	19	Funcionamiento del detector, generador de histograma	Integral	Enviar imágenes desde la interfaz para verificar el correcto procesamiento de imágenes.	El histograma de la imagen enviada y el histograma reportado por el instrumento deben coincidir.	FFGA Vivado Vivado SDK Interfaz	15/10/21	-	Acordé a lo esperado
	20	Captura y procesamiento de eventos	Homologación	1- Capturar 60 imágenes antes de irradiar, calcular el umbral en base a la gaussiana del ruido. 2- Capturar 60 imágenes con la piedra de americio sobre el sensor. 3- Capturar 60 imágenes después de irradiar para ver el estado del sensor. Realizar esto para ganancia x1.5, x8 y x1 en ese orden.	Observar que se forma una gaussiana y a medida que la ganancia disminuye, se corre hacia la izquierda. Observar que la cantidad de píxeles muertos después de irradiar sea mayor a antes de irradiar.	FFGA Vivado Vivado SDK Interfaz Piedra de americio Contenedor oscuro	06/11/21	Para la generación de eventos se utilizó una piedra de Am241	Acordé a lo esperado
SELFIE	21	Captura y procesamiento de eventos para partículas gamma	Homologación	1- Cubrir el sensor con una servilleta y colocar la piedra de americio sobre ella. 2- Configurar la ganancia en x1.5. 3- Dejar captuando al instrumento por 24 horas y construir histograma.	Se debe obtener un histograma de intensidades con una concentración de eventos para intensidades bajas.	FFGA Vivado Vivado SDK Interfaz Piedra de americio Contenedor oscuro	13/11/21	Para la generación de eventos se utilizó una piedra de Am241	No se pudo finalizar. Los archivos del SDK de Xilinx se corrompieron a las 3 horas aproximadamente.